

Fundamentos de bases de datos

Tarea 3

Modelo Relacional

Díaz Gómez Silvia
Eugenio Aceves Narciso Isaac
Quiroz Castañeda Edgar

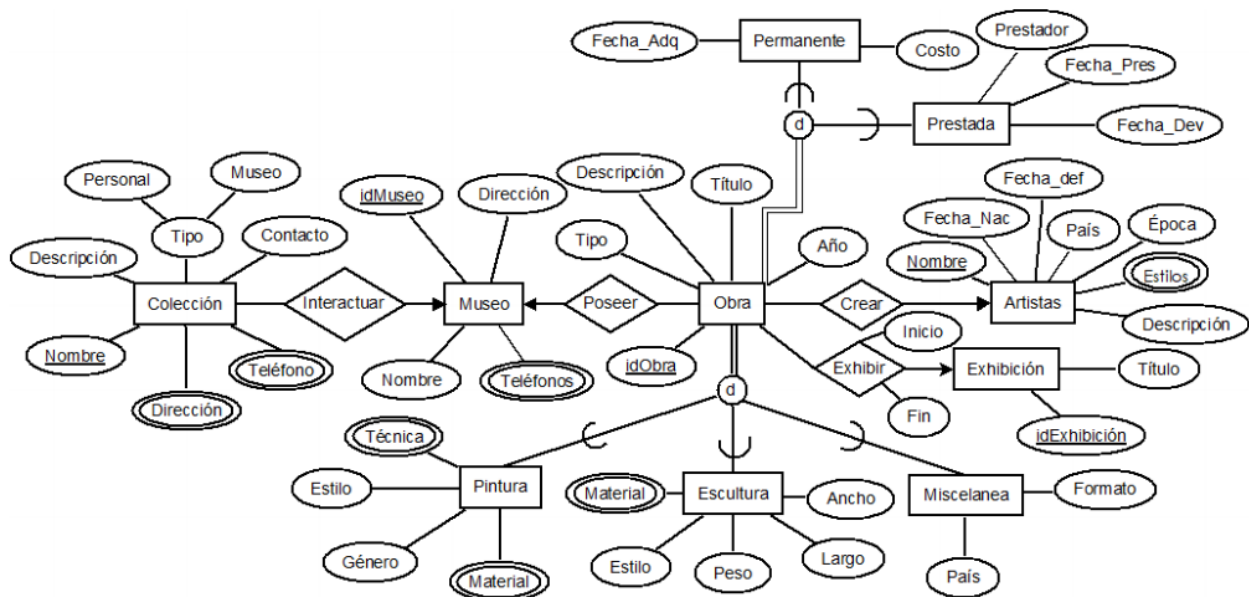
22 de marzo del 2019

1. Preguntas de repaso

- ¿Qué es una **relación** y qué características tiene?
- ¿Qué es un **esquema de relación**?
- ¿Qué es una **llave primaria** ¿qué es una **llave candidata**? ¿qué es una **llave mínima**? ¿qué es una **super llave**?
- ¿Qué restricciones impone una **llave primaria** y una llave foránea al modelo de dato relacional?
- Investiga cómo se traducen las **categorías** (presentes en el **modelos E/R**) al **modelos relacional**. Proporciona un ejemplo.

2. Modelo relacional

Traduce el siguiente modelo **Entidad/Relación** a su correspondiente **Modelo Relacional**:



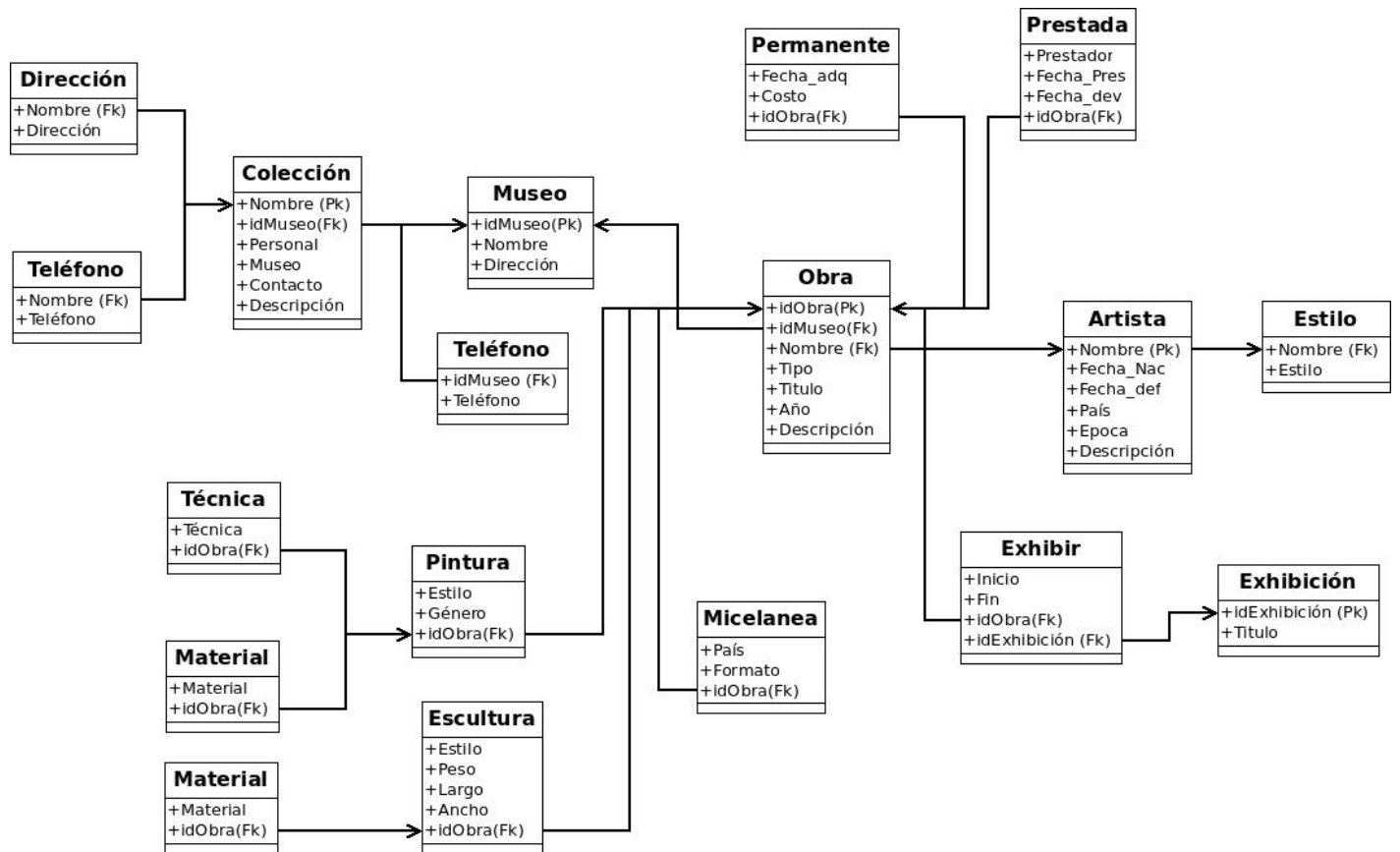


Figura 1: Traducción del modelo E-R de la figura anterior al modelo Relacional.

3. Modelo relacional

Traduce s su correspondiente **Modelo Relacional** el problema del **Sistema de Información Geográfica (Tarea 1)**. Se realizaste alguna modificación a tu diseño original (para mejorarlo) indica los cambios hechos y la justificación de los mismos.

En cualquier caso, deberás mostrar el **diagrama E/R** y su correspondiente traducción. Es importante que muetres tanto las **restricciones de entidad** como las de **integridad referencial**.

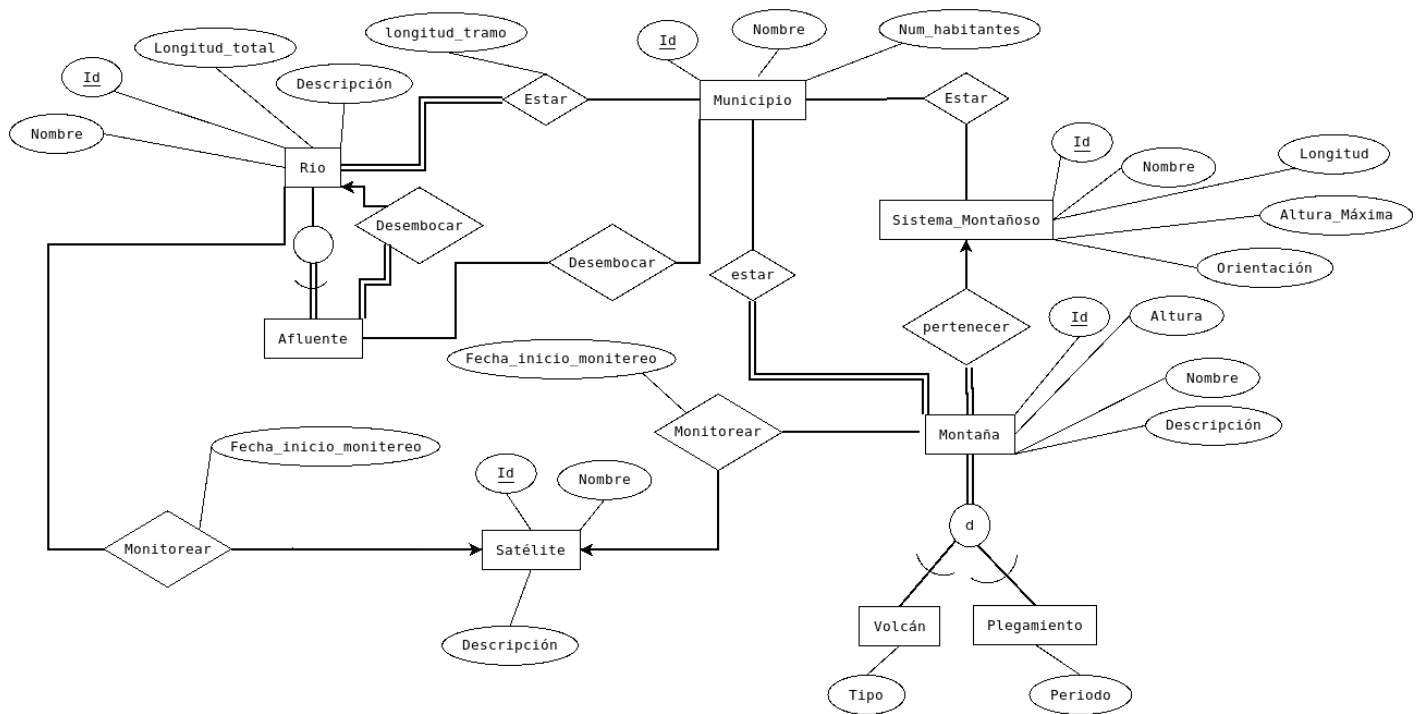


Figura 2: Modelo E-R para el Sistema de Información Geográfica.

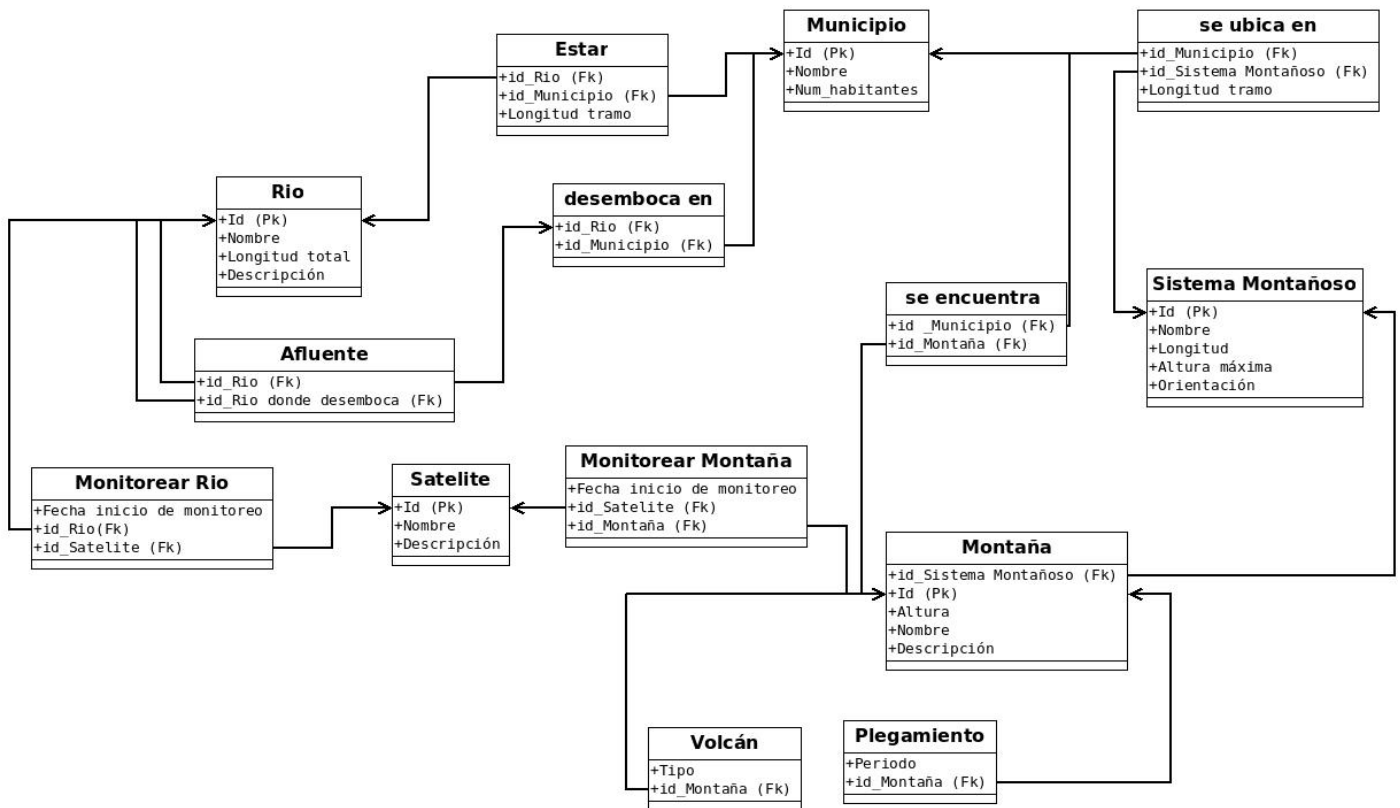


Figura 3: Traducción del modelo E-R para el Sistema de Información Geográfica a Esquema Relacional.

4. Lectura

Leer el artículo **Codd's 12 Rules for a RDBMS**. Explica con tus propias palabras cada una de las 12 reglas de Codd.

Indica por qué consideras que son importantes y si, hasta el momento de lo comentado en el curso, sería posible que un SDBD pudiera cumplir enteramente con lo que ahí se propone.

- **Regla 1 : The Information Rule**

Toda la información debe estar representada en el esquema lógico de la base de datos. Y debe estar representada como entradas de una tabla.

Si los datos no están en el esquema lógico, entonces no sería accesibles usando los mecanismos de consulta del sistema, por lo que es como si los datos no estuvieran ahí.

Esto se puede lograr en SDBD forzando a que la única manera de almacenar datos sea a través de inserciones en tablas.

- **Regla 2 : Guaranteed Access Rule** Toda la información debe ser accesible. Esto es que todo dato tenga un identificador o llave primaria, además de que los datos sean atómicos ya que son de suma importancia para poder garantizar la accesibilidad a los datos.

Esto es importante porque no poder acceder a los datos es equivalente a no tener los datos.

Sería posible que un SDBD pueda garantizar esto si a todos los datos insertados bien se les exige una llave al momento o se les asigna un llave sintética al ser ingresados.

Aunque forzar lo segundo podría llevar bien a redundancia o que ya no se refleje totalmente el esquema lógico de la base de datos.

- **Regla 3: Systematic Treatment of NULL Values**

Debe existir un valor NULL diferente de todos los demás posibles datos que represente la ausencia de información y éste debe ser tratado de la misma manera que todos los demás datos por el sistema.

Esto es importante porque en otro caso no se podría representar la ausencia de información.

Esto se puede lograr en un SDBD simplemente añadiendo el caso.

- **Regla 4: Dynamic Online Catalog Based on the Relational Model**

La estructura de la base de datos debe estar almacenada de igual manera que el resto de los datos, para que pueda ser accesada por usuarios usando los mecanismos ordinarios de consulta.

- **Regla 5: Comprehensive Data Sublanguage Rule**

Se debe tener soporte para algún lenguaje formal que permita definir tipos de datos, hacer consultas, transacciones, y definir restricciones.

- **Regla 6: View Updating Rule**

Todas las vistas que son teóricamente actualizables deben de poder ser actualizadas.

- **Regla 7: High-Level Insert, Update, and Delete**

Las operaciones de modificación de la base de datos deben estar disponibles en términos de conjuntos de tuplas, no de tuplas individuales.

- **Regla 8: Physical Data Independence**

Esta regla menciona que a nivel físico que es donde la base de datos almacena e implementa los métodos de acceso a los datos es independiente de la manera lógica en que se accede, por lo tanto los cambios que se hagan a nivel físico no le afecta al usuario puesto que al usuario no le interesa saber como se almacena o como se acceden a los datos.

- **Regla 9: Logical Data Independence**

El esquema de la base de datos debe ser independiente de las aplicaciones que utilizan la base de datos. Esto es que se pueda modificar la estructura lógica de la base de datos sin necesidad de modificar las aplicaciones que la utilizan.

- **Regla 10: Integrity Independence**

Las restricciones de integridad deben de ser independientes del funcionamiento de aplicaciones. Esto es que esas restricciones se puedan modificar sin necesidad de modificar las aplicaciones que utilizan la base de datos.

- **Regla 11: Distribution Independence**

Una base de datos distribuida debe ser idéntica a una no distribuida desde el punto de vista del usuario.

- **Regla 12: Non-Subversion Rule**

Indica que no debe existir un mecanismo para violar las restricciones. Es decir, en caso de que se proporcione acceso de bajo nivel al sistema, esta interfaz no debe ser capaz de modificar sin restricciones.